

# Análisis Estadístico de Resultados

## Contents

<b>ANÁLISIS DE CONJUNTOS DE ENTRENAMIENTO</b>	<b>1</b>
ANÁLISIS DE VALORES DE BRISQUE . . . . .	1
ANÁLISIS DE VALORES DE NIQE . . . . .	3
ANÁLISIS DE VALORES DE PIQE . . . . .	5
<b>ANÁLISIS DE CONJUNTOS DE VALIDACIÓN</b>	<b>7</b>
ANÁLISIS DE VALORES DE BRISQUE . . . . .	7
ANÁLISIS DE VALORES DE NIQE . . . . .	9
ANÁLISIS DE VALORES DE PIQE . . . . .	11
ANÁLISIS DEL CLASIFICADOR ENTRENADO (VERSIÓN IMT: 87% de precisión y con 88% de especificad) . . . . .	13
ANÁLISIS DEL CLASIFICADOR ENTRENADO (VERSIÓN X: precisión de 80% y especificad de 97%) . . . . .	15

```
#install.packages('dplyr', dep=TRUE)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

setwd("/home/joseluis/Dropbox/STATIC/Docencia/TFG/TFG_JoseMontero/medidas_calidad")
```

## ANÁLISIS DE CONJUNTOS DE ENTRENAMIENTO

```
dataBuenaCalidad=read.csv("MetricasBuenaCalidadEntreno.csv", stringsAsFactors=FALSE)
dataMalaCalidad=read.csv("MetricasMalaCalidadEntreno.csv", stringsAsFactors=FALSE)
```

## ANÁLISIS DE VALORES DE BRISQUE

### DIAGRAMAS DE CAJA DE LOS VALORES DE BRISQUE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
brisqueB=dataBuenaCalidad$BRISQUE_SCORE
brisqueM=dataMalaCalidad$BRISQUE_SCORE

dfB=data.frame(brisqueB)
dfB['Calidad']='Buena'
```

```

names(dfB)[names(dfB) == 'brisqueB'] = 'Brisque'

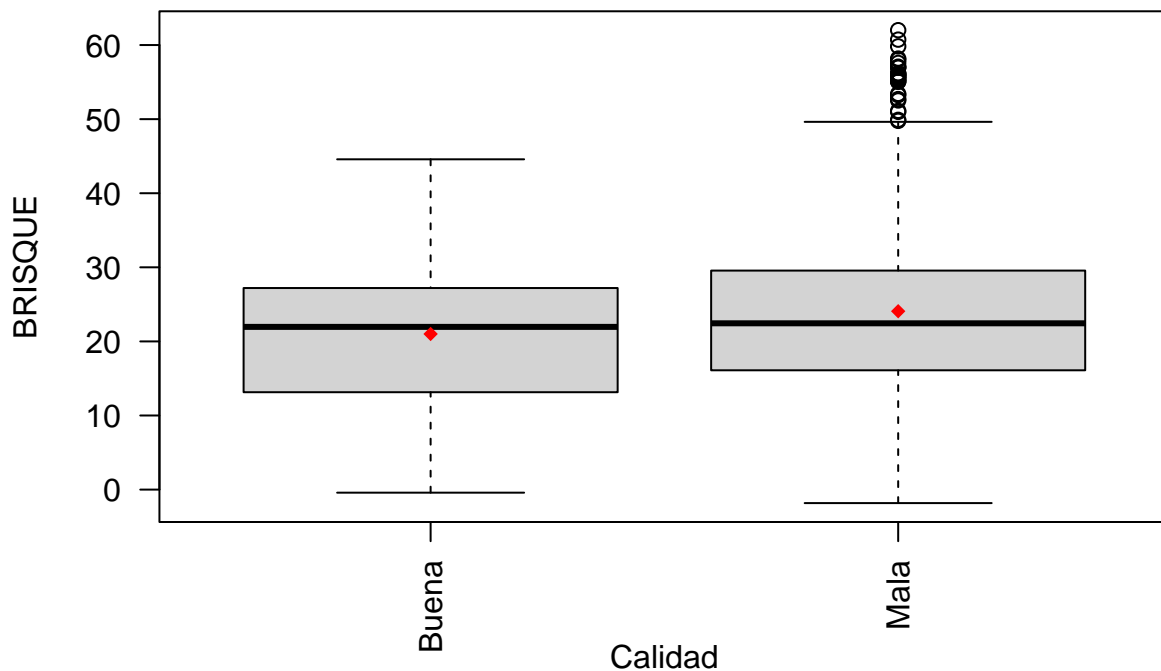
dfM=data.frame(brisqueM)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'brisqueM'] = 'Brisque'

dfBrisque=bind_rows(dfB, dfM)

#dfBrisque[1,]
#dfBrisque[1000,]

medias=aggregate(Brisque~Calidad, data=dfBrisque, FUN=mean)
boxplot(Brisque~Calidad, data=dfBrisque, ylab="BRISQUE", las=2)
points(medias, col="red", pch=18)

```



**TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE BRISQUE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD**

```

var.test(brisqueB, brisqueM, alternative="two.sided", conf.level=0.95)

##
## F test to compare two variances
##
## data: brisqueB and brisqueM
## F = 0.7307, num df = 600, denom df = 800, p-value = 4.815e-05

```

```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6295432 0.8494905
## sample estimates:
## ratio of variances
##          0.7307001
```

La conclusión del test es que no se puede considerar que las varianzas sean iguales.

## TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE BRISQUE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(brisqueB, brisqueM, alternative="less", mu=0, paired=FALSE, var.equal=FALSE, conf.level=0.95)

##
## Welch Two Sample t-test
##
## data: brisqueB and brisqueM
## t = -5.3352, df = 1376.4, p-value = 5.574e-08
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -2.12448
## sample estimates:
## mean of x mean of y
##  21.00271  24.07503
```

La conclusión del test es que no se puede considerar que las medias sean iguales. Aceptamos que la media de los valores para las imágenes de buena calidad es inferior a la media para las imágenes de mala calidad.

Aún así, los diagramas de caja indican que para muchas imágenes el valor de BRISQUE no mide correctamente su nivel de calidad.

## ANÁLISIS DE VALORES DE NIQE

### DIAGRAMAS DE CAJA DE LOS VALORES DE NIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
niqeB=dataBuenaCalidad$NIQE_SCORE
niqeM=dataMalaCalidad$NIQE_SCORE

dfB=data.frame(niqeB)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'niqeB'] = 'Niqe'

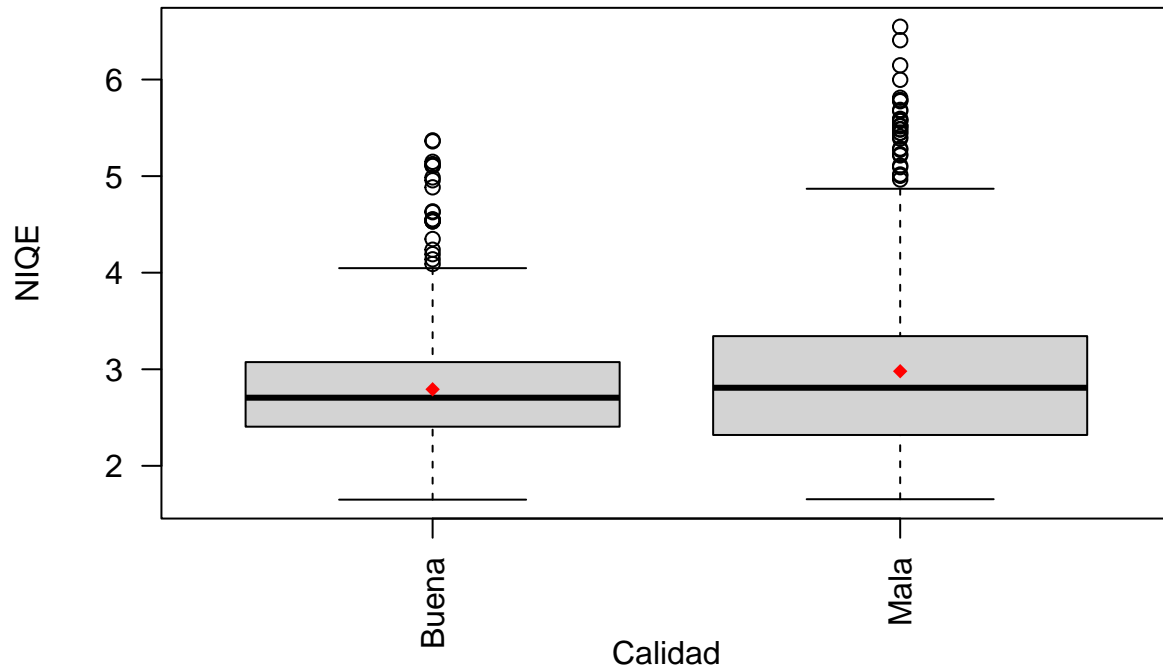
dfM=data.frame(niqeM)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'niqeM'] = 'Niqe'

dfNiqe=bind_rows(dfB, dfM)

#dfNiqe[1,]
#dfNiqe[1000,]

medias=aggregate(Niqe~Calidad, data=dfNiqe, FUN=mean)
```

```
boxplot(Niqe~Calidad, data=dfNiqe, ylab="NIQE", las=2)
points(medias, col="red", pch=18)
```



### TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE NIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(niqeB, niqeM, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: niqeB and niqeM
## F = 0.49223, num df = 600, denom df = 800, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.4240880 0.5722542
## sample estimates:
## ratio of variances
## 0.4922318
```

La conclusión del test es que no se puede considerar que las varianzas sean iguales.

## TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE NIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(niqeB, niqeM, alternative="less", mu=0, paired=FALSE, var.equal=FALSE, conf.level=0.95)

##
## Welch Two Sample t-test
##
## data: niqeB and niqeM
## t = -4.7148, df = 1394, p-value = 1.331e-06
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.1217295
## sample estimates:
## mean of x mean of y
##  2.792714  2.979731
```

La conclusión del test es que no se puede considerar que las medias sean iguales. Aceptamos que la media de los valores para las imágenes de buena calidad es inferior a la media para las imágenes de mala calidad.

Aún así, los diagramas de caja indican que para muchas imágenes el valor de NIQE no mide correctamente su nivel de calidad.

## ANÁLISIS DE VALORES DE PIQE

### DIAGRAMAS DE CAJA DE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
piqeB=dataBuenaCalidad$PIQE_SCORE
piqeM=dataMalaCalidad$PIQE_SCORE

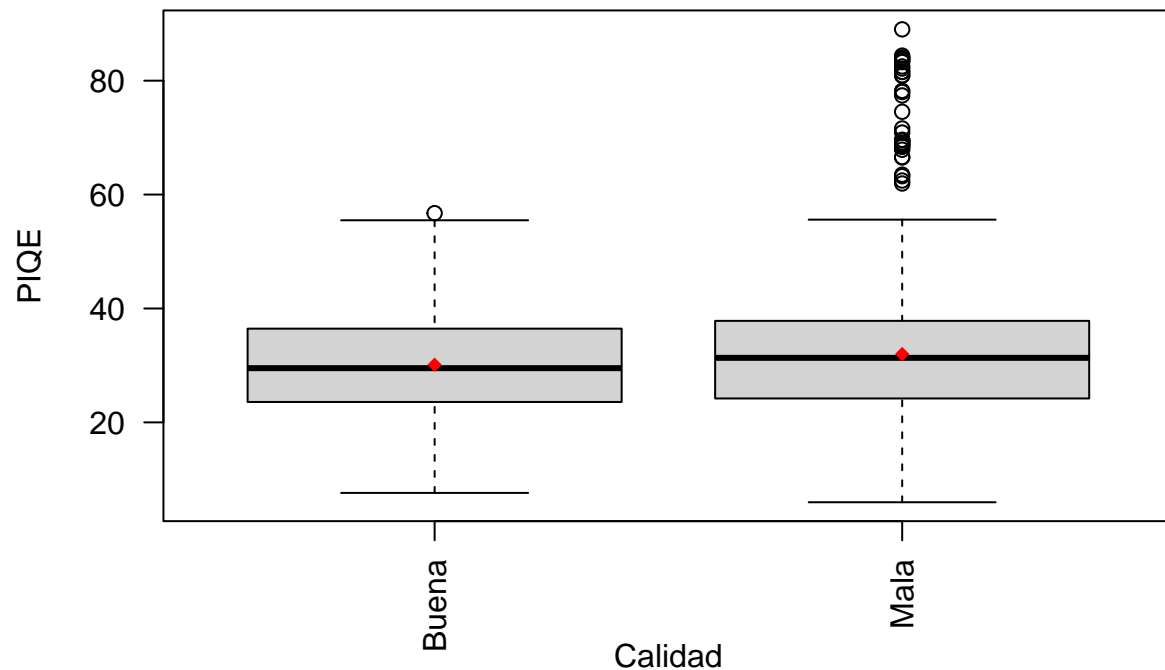
dfB=data.frame(piqeB)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'piqeB'] = 'Piqe'

dfM=data.frame(piqeM)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'piqeM'] = 'Piqe'

dfPiqe=bind_rows(dfB, dfM)

#dfPiqe[1,]
#dfPiqe[1000,]

medias=aggregate(Piqe~Calidad, data=dfPiqe, FUN=mean)
boxplot(Piqe~Calidad, data=dfPiqe, ylab="PIQE", las=2)
points(medias, col="red", pch=18)
```



### TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(piqeB, piqeM, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: piqeB and piqeM
## F = 0.49641, num df = 600, denom df = 800, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.4276864 0.5771097
## sample estimates:
## ratio of variances
##      0.4964083
```

La conclusión del test es que no se puede considerar que las varianzas sean iguales.

### TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(piqeB, piqeM, alternative="less", mu=0, paired=FALSE, var.equal=FALSE, conf.level=0.95)
```

```
##
## Welch Two Sample t-test
```

```
##
## data:  piqueB and piqueM
## t = -2.9966, df = 1394.7, p-value = 0.001389
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.8470649
## sample estimates:
## mean of x mean of y
##  30.11980  31.99915
```

La conclusión del test es que no se puede considerar que las medias sean iguales. Aceptamos que la media de los valores para las imágenes de buena calidad es inferior a la media para las imágenes de mala calidad.

Aún así, los diagramas de caja indican que para muchas imágenes el valor de PIQE no mide correctamente su nivel de calidad.

## ANÁLISIS DE CONJUNTOS DE VALIDACIÓN

```
dataBuenaCalidad=read.csv("MetricasBuenaCalidadValidacion.csv", stringsAsFactors=FALSE)
dataMalaCalidad=read.csv("MetricasMalaCalidadValidacion.csv", stringsAsFactors=FALSE)
```

## ANÁLISIS DE VALORES DE BRISQUE

### DIAGRAMAS DE CAJA DE LOS VALORES DE BRISQUE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
brisqueB=dataBuenaCalidad$BRISQUE_SCORE
brisqueM=dataMalaCalidad$BRISQUE_SCORE

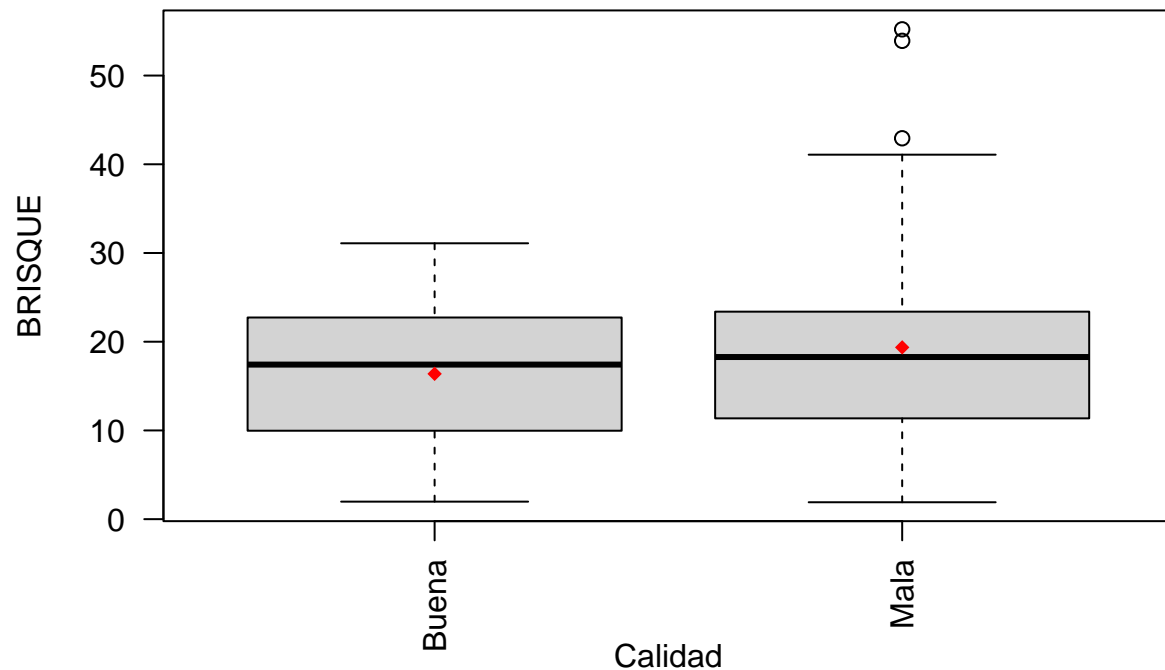
dfB=data.frame(brisqueB)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'brisqueB'] = 'Brisque'

dfM=data.frame(brisqueM)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'brisqueM'] = 'Brisque'

dfBrisque=bind_rows(dfB, dfM)

#dfBrisque[1,]
#dfBrisque[1000,]

medias=aggregate(Brisque~Calidad, data=dfBrisque, FUN=mean)
boxplot(Brisque~Calidad, data=dfBrisque, ylab="BRISQUE", las=2)
points(medias, col="red", pch=18)
```



### TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE BRISQUE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(brisqueB, brisqueM, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: brisqueB and brisqueM
## F = 0.46356, num df = 38, denom df = 58, p-value = 0.01327
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.2627054 0.8480621
## sample estimates:
## ratio of variances
## 0.4635648
```

La conclusión del test es que no se puede considerar que las varianzas sean iguales.

### TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE BRISQUE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(brisqueB, brisqueM, alternative="less", mu=0, paired=FALSE, var.equal=FALSE, conf.level=0.95)
```

```
##
## Welch Two Sample t-test
```



```
##
## data:  brisqueB and brisqueM
## t = -1.5975, df = 95.892, p-value = 0.05672
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.1185586
## sample estimates:
## mean of x mean of y
## 16.37794 19.36421
```

La conclusión del test es que no se puede considerar que las medias sean diferentes. Aceptamos que la media de los valores para las imágenes de buena calidad es **IGUAL** a la media para las imágenes de mala calidad.

## ANÁLISIS DE VALORES DE NIQE

### DIAGRAMAS DE CAJA DE LOS VALORES DE NIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
niqeB=dataBuenaCalidad$NIQE_SCORE
niqeM=dataMalaCalidad$NIQE_SCORE

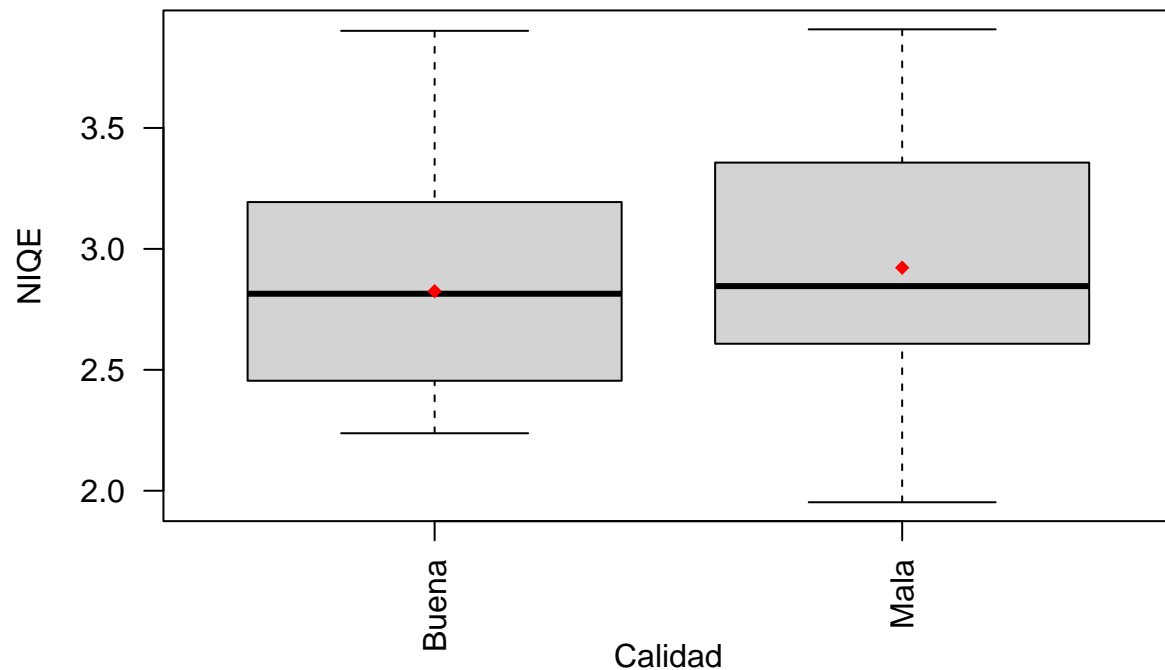
dfB=data.frame(niqeB)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'niqeB'] = 'Niqe'

dfM=data.frame(niqeM)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'niqeM'] = 'Niqe'

dfNiqe=bind_rows(dfB, dfM)

#dfNiqe[1,]
#dfNiqe[1000,]

medias=aggregate(Niqe~Calidad, data=dfNiqe, FUN=mean)
boxplot(Niqe~Calidad, data=dfNiqe, ylab="NIQE", las=2)
points(medias, col="red", pch=18)
```



### TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE NIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(niqeB, niqeM, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: niqeB and niqeM
## F = 0.59814, num df = 38, denom df = 58, p-value = 0.09437
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3389708 1.0942613
## sample estimates:
## ratio of variances
## 0.5981414
```

La conclusión del test es que no se puede considerar que las varianzas sean diferentes

### TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE NIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(niqeB, niqeM, alternative="less", mu=0, paired=FALSE, var.equal=TRUE, conf.level=0.95)
```

```
##
## Two Sample t-test
```

```
##
## data:  niqeB and niqeM
## t = -0.90496, df = 96, p-value = 0.1839
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.08111631
## sample estimates:
## mean of x mean of y
##  2.825029  2.922140
```

La conclusión del test es que no se puede considerar que las medias sean diferentes. Aceptamos que la media de los valores para las imágenes de buena calidad es **IGUAL** a la media para las imágenes de mala calidad.

## ANÁLISIS DE VALORES DE PIQE

### DIAGRAMAS DE CAJA DE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
piqeB=dataBuenaCalidad$PIQE_SCORE
piqeM=dataMalaCalidad$PIQE_SCORE

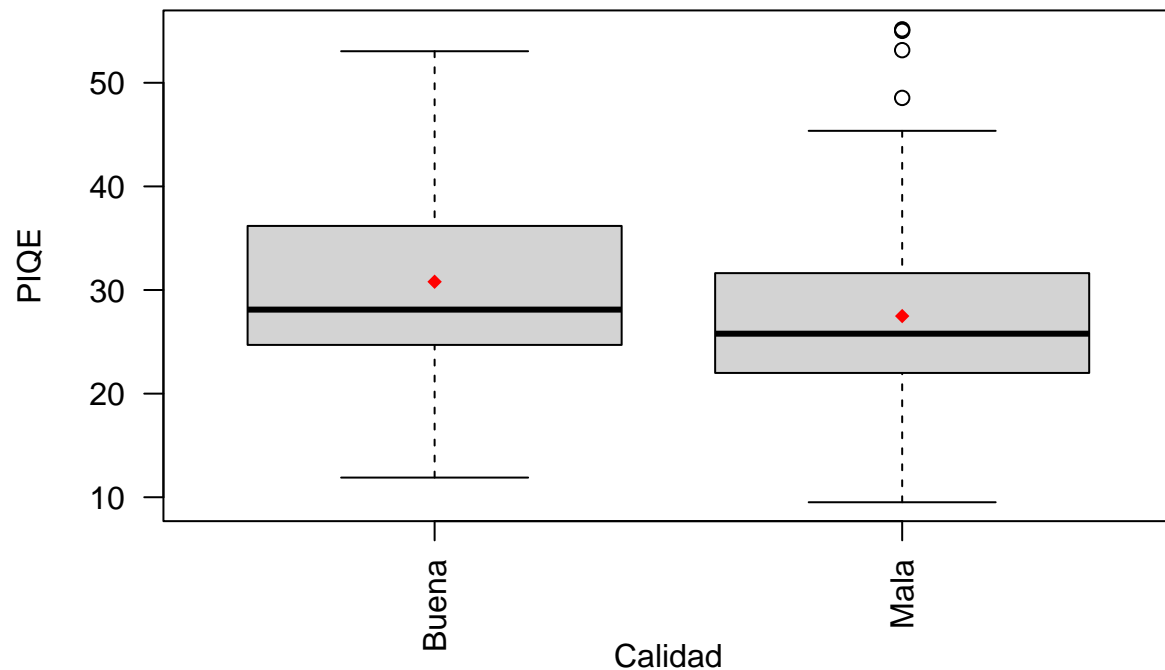
dfB=data.frame(piqeB)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'piqeB'] = 'Piqe'

dfM=data.frame(piqeM)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'piqeM'] = 'Piqe'

dfPiqe=bind_rows(dfB, dfM)

#dfPiqe[1,]
#dfPiqe[1000,]

medias=aggregate(Piqe~Calidad, data=dfPiqe, FUN=mean)
boxplot(Piqe~Calidad, data=dfPiqe, ylab="PIQE", las=2)
points(medias, col="red", pch=18)
```



### TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(piqeB, piqeM, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: piqeB and piqeM
## F = 1.2819, num df = 38, denom df = 58, p-value = 0.3877
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7264615 2.3451535
## sample estimates:
## ratio of variances
##          1.2819
```

La conclusión del test es que no se puede considerar que las varianzas sean diferentes.

### TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(piqeB, piqeM, alternative="two.sided", mu=0, paired=FALSE, var.equal=TRUE, conf.level=0.95)
```

```
##
## Two Sample t-test
```

```
##
## data:  piqueB and piqueM
## t = 1.5664, df = 96, p-value = 0.1206
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8852272  7.5097674
## sample estimates:
## mean of x mean of y
##  30.79825  27.48598
```

La conclusión del test es que no se puede considerar que las medias sean diferentes. Aceptamos que la media de los valores para las imágenes de buena calidad es **IGUAL** a la media para las imágenes de mala calidad.

## ANÁLISIS DEL CLASIFICADOR ENTRENADO (VERSIÓN IMT: 87% de precisión y con 88% de especificad)

```
dataBuenaCalidad=read.csv("Metricas_Buenas_imagenes_Val_Modelos.csv", stringsAsFactors=FALSE)
dataMalaCalidad=read.csv("Metricas_malas_imagenes_Val_Modelos.csv", stringsAsFactors=FALSE)
```

## DIAGRAMAS DE CAJA DE LOS SCORES PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
scoreB1=dataBuenaCalidad$X..Mala.calidad.Modelo.IMT
scoreM1=dataMalaCalidad$X..Mala.calidad.Modelo.IMT

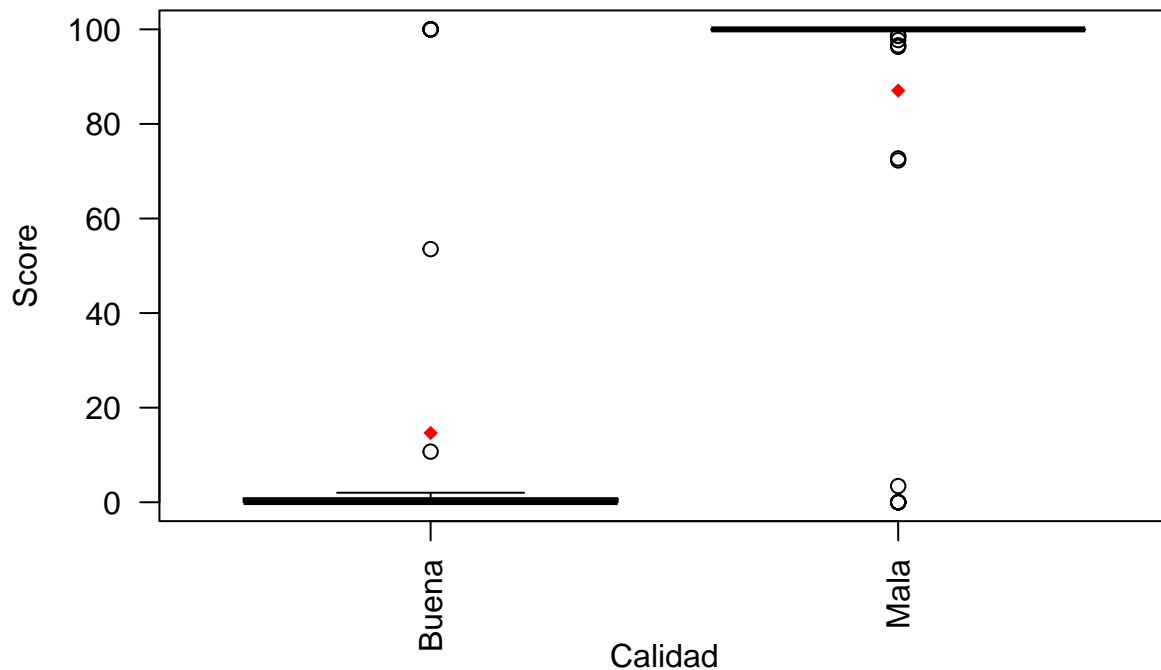
dfB=data.frame(scoreB1)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'scoreB1'] = 'Score'

dfM=data.frame(scoreM1)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'scoreM1'] = 'Score'

dfScore=bind_rows(dfB, dfM)

#dfScore[1,]
#dfScore[50,]

medias=aggregate(Score~Calidad, data=dfScore, FUN=mean)
boxplot(Score~Calidad, data=dfScore, ylab="Score", las=2)
points(medias, col="red", pch=18)
```



## TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE SCORE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(scoreB1, scoreM1, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: scoreB1 and scoreM1
## F = 1.1168, num df = 38, denom df = 58, p-value = 0.6935
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.6328751 2.0430394
## sample estimates:
## ratio of variances
## 1.116759
```

La conclusión del test es que no se puede considerar que las varianzas sean diferentes.

### TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(scoreB1, scoreM1, alternative="less", mu=0, paired=FALSE, var.equal=TRUE, conf.level=0.95)
```

```
##  
## Two Sample t-test
```

```
##
## data:  scoreB1 and scoreM1
## t = -10.579, df = 96, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -61.04021
## sample estimates:
## mean of x mean of y
##  14.64077  87.04878
```

La conclusión del test es que no se puede considerar que las medias sean iguales. Aceptamos que la media de los valores para las imágenes de buena calidad es **DIFERENTE** a la media para las imágenes de mala calidad.

## ANÁLISIS DEL CLASIFICADOR ENTRENADO (VERSIÓN X: precisión de 80% y especificidad de 97%)

```
dataBuenaCalidad=read.csv("Metricas_Buenas_imagenes_Val_Modelos.csv", stringsAsFactors=FALSE)
dataMalaCalidad=read.csv("Metricas_malas_imagenes_Val_Modelos.csv", stringsAsFactors=FALSE)
```

## DIAGRAMAS DE CAJA DE LOS SCORES PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
scoreB2=dataBuenaCalidad$X..Mala.calidad.Modelo.X
scoreM2=dataMalaCalidad$X..Mala.calidad.Modelo.X

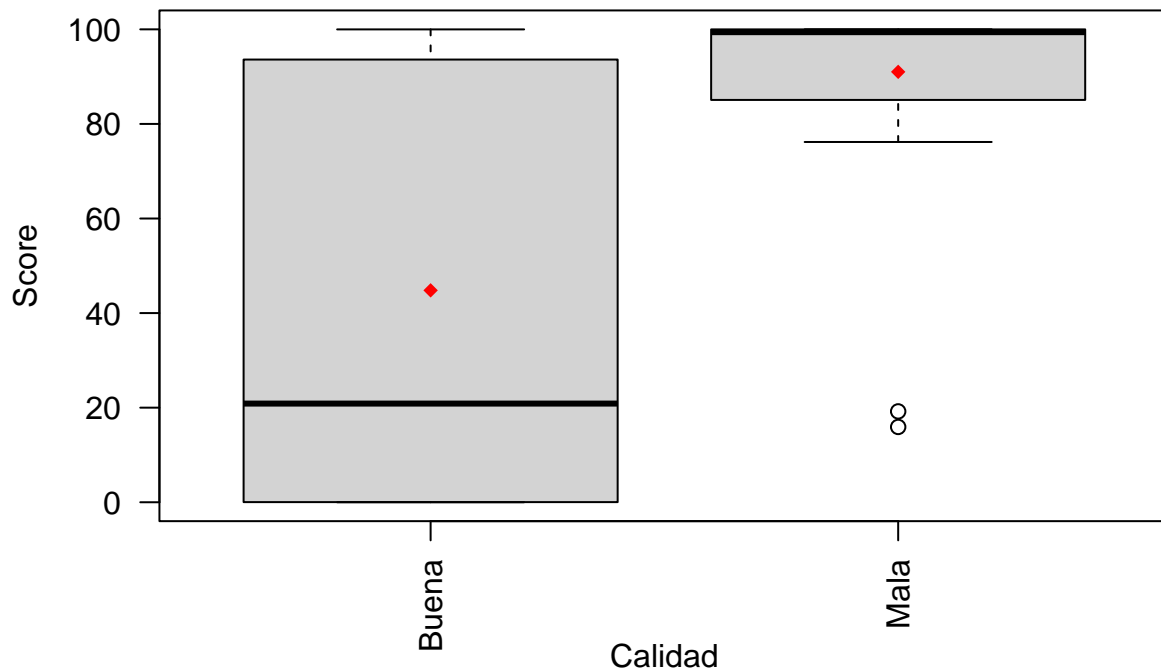
dfB=data.frame(scoreB2)
dfB['Calidad']='Buena'
names(dfB)[names(dfB) == 'scoreB2'] = 'Score'

dfM=data.frame(scoreM2)
dfM['Calidad']='Mala'
names(dfM)[names(dfM) == 'scoreM2'] = 'Score'

dfScore=bind_rows(dfB, dfM)

#dfScore[1,]
#dfScore[50,]

medias=aggregate(Score~Calidad, data=dfScore, FUN=mean)
boxplot(Score~Calidad, data=dfScore, ylab="Score", las=2)
points(medias, col="red", pch=18)
```



### TEST DE IGUALDAD DE VARIANZAS ENTRE LOS VALORES DE SCORE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
var.test(scoreB2, scoreM2, alternative="two.sided", conf.level=0.95)
```

```
##
## F test to compare two variances
##
## data: scoreB2 and scoreM2
## F = 7.935, num df = 38, denom df = 58, p-value = 4.584e-12
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  4.496818 14.516569
## sample estimates:
## ratio of variances
##      7.934998
```

La conclusión del test es que no se puede considerar que las varianzas sean iguales.

### TEST DE IGUALDAD DE MEDIAS ENTRE LOS VALORES DE PIQE PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD

```
t.test(scoreB2, scoreM2, alternative="less", mu=0, paired=FALSE, var.equal=FALSE, conf.level=0.95)
```

```
##
## Welch Two Sample t-test
```



```
##
## data: scoreB2 and scoreM2
## t = -6.107, df = 44.393, p-value = 1.137e-07
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -33.49371
## sample estimates:
## mean of x mean of y
##  44.80811  91.01149
```

La conclusión del test es que no se puede considerar que las medias sean iguales. Aceptamos que la media de los valores para las imágenes de buena calidad es **DIFERENTE** a la media para las imágenes de mala calidad.

Aún así, los diagramas de caja indican que para muchas imágenes el valor de Score no mide correctamente su nivel de calidad.

**Conclusión:** El Modelo IMT permite una mejor separación de las imágenes de buena y mala calidad.

## DIAGRAMAS DE CAJA DE LOS SCORES PARA LAS IMÁGENES DE BUENA Y MALA CALIDAD DE LOS DOS MODELOS

```
dfB1=data.frame(scoreB1)
dfB1['Calidad']='A: BuenaIMT'
names(dfB1)[names(dfB1) == 'scoreB1'] = 'Score'

dfM1=data.frame(scoreM1)
dfM1['Calidad']='B: MalaIMT'
names(dfM1)[names(dfM1) == 'scoreM1'] = 'Score'

dfB2=data.frame(scoreB2)
dfB2['Calidad']='C: BuenaX'
names(dfB2)[names(dfB2) == 'scoreB2'] = 'Score'

dfM2=data.frame(scoreM2)
dfM2['Calidad']='D: MalaX'
names(dfM2)[names(dfM2) == 'scoreM2'] = 'Score'

dfScore=bind_rows(dfB1, dfM1, dfB2, dfM2)

medias=aggregate(Score~Calidad, data=dfScore, FUN=mean)
boxplot(Score~Calidad, data=dfScore, ylab="Score", las=2)
points(medias, col="red", pch=18)
```

